ON CONSTRUCTING THREE DIMENSIONAL OVERLAPPING GRIDS WITH CMPGRD

William D. Henshaw
Geoffrey Chesshire
Michael E. Henderson
IBM Research Division
Thomas J. Watson Research Centre
Yorktown Heights, NY 10598

ABSTRACT

We describe some techniques for the construction of three-dimensional composite overlapping grids using the grid construction program CMPGRD. The overlapping approach can be used to generate grids for regions of complicated geometry. The grids can be constructed to be smooth and free from coordinate singularities. The ability to create smooth grids for complicated regions is an important first step towards the accurate numerical solution of partial differential equations. We describe how to create grids for surfaces defined by cross-sections such as an airplane wing. We also describe how the patched surfaces generated from a CAD package can be used within the CMPGRD program and how grids can be created in regions where surfaces intersect.

INTRODUCTION

We describe some techniques for the construction of three-dimensional composite overlapping grids using the grid construction program CMPGRD. A composite overlapping grid consists of a set of logically rectangular component grids which cover a region and overlap where they meet. Interpolation conditions connect solution values defined on the grid. In the paper *Composite Meshes for the Solution of Partial Differential Equations* [1] we described CMPGRD and discussed techniques for the solution of elliptic and hyperbolic partial differential equations (PDEs) on overlapping grids. Our examples were limited to two-dimensional problems. In this paper we emphasize the extensions made to the code for the generation of three-dimensional grids. The overlapping approach can be used to generate grids for regions of complicated geometry. The region can be divided into a number of sub-domains for which a component grid can be more easily created. The component grids can be constructed to be smooth and free from coordinate singularities. Once all component grids have been created the CMPGRD program will automatically determine the interpolation conditions which connect the grids. CMPGRD has a very general algorithm for doing this which supports any number of component grids overlapping in any order. CMPGRD can generate the overlapping grids appropriate for higher order interpolation, higher order discretization, cell-centred or cell-vertex grids and the sequence of grids which can be used for the multigrid algorithm [2]. Some other references which describe overlapping grids and the solution of PDEs thereon include [3] [4] [5] [6] [7].

The problem of grid construction can be made simpler by using the composite overlapping grid technique. However, the task of creating component grids, especially in three-dimensions, can still be very difficult. In this paper we discuss some ways to create three-dimensional component grids suitable for use with CMPGRD. The emphasis here is on techniques which are closely linked to the overlapping approach and to methods which avoid coordinate singularities.

We first describe how to create grids for surfaces defined by cross-sections, such as an ellipsoid or wing, where the cross-sections may converge to a point at one or both ends. The basic underlying problem here is the creation of a composite grid about a sphere which is free from singularities. A solution to this problem is to cover the sphere with more than one patch. We use an orthographic projection to create grids over the north and south poles of the sphere. This approach for a sphere can be generalized to more general surfaces which are defined by cross-sections: a separate patch is placed on the end of the surface where the cross-sections converge to a point.

A complicated object such as an airplane consists of multiple surfaces which connect to each other, such as when a wing joins a fuselage. Using the cross-section technique component grids can be created for the wing and fuselage but there remains the problem of connecting the wing to the fuselage. The connection may either be smooth, in which case a fillet grid is appropriate, or the connection may be a corner in which case the curve of intersection between the two surfaces should be an edge of a grid. We consider the latter case here. One way to create a grid for the wing which matches to the fuselage is to project the end face of the wing grid onto the surface of the fuselage. We present an example of this technique. We have developed another, more general, approach for creating grids in the region where surfaces intersect. This method begins by computing the curve of intersection between the surfaces and then reparameterizing the surface using the intersecting curves to define the portion of the surface to use. We give several examples to illustrate this approach.

For the description of complicated three-dimensional regions there are many advantages to using a computer aided design (CAD) package. We show how the patched surfaces generated from a CAD package can be used within the CMPGRD program. Although the patched surface may be smooth, the parameterization of the surface may not be smooth and it is thus necessary to reparameterize the surface. These reparameterized surfaces can then be used in CMPGRD to create overlapping grids. We show an example of a grid for a wing connected to an engine nacelle by a pylon.

CREATING THREE-DIMENSIONAL OVERLAPPING GRIDS

A composite overlapping grid consists of a set of component grids. Each component grid covers a portion of the computational region. Component grids overlap where they meet. Solution values are matched by interpolation at the overlapping boundary. In order to determine how to interpolate between grids CMPGRD requires knowledge of the component grid mapping not only at grid points but also at all intermediate positions. Thus the component grid must be defined as a continuous mapping.

Component grid mapping: For the purposes of CMPGRD a component grid is defined as a smooth mapping from a unit cube, $\mathbf{r}$, to the computational domain $\mathbf{x} \in R^3$. CMPGRD needs to evaluate this mapping at any point $\mathbf{r}$ in the unit cube and requires both the image of the mapping, $\mathbf{x}(\mathbf{r})$, and its derivatives $\partial\mathbf{x}(\mathbf{r})/\partial\mathbf{r}$:

$$\text{Component Grid Mapping}: \mathbf{r} \rightarrow (\mathbf{x}, \frac{\partial \mathbf{x}}{\partial \mathbf{r}}).$$

A component grid has a number of characteristics associated with it. For example, each face of the component grid has a boundary condition. The face may be part of a true boundary, or the grid may be periodic, or the derivative of the grid may be periodic or the face may be used for interpolation.

CMPGRD provides various features for creating two-dimensional and three-dimensional component grids. One approach is for the user to provide a complete description of the component grid which is supplied to CMPGRD as an *externally* defined grid (i.e. Fortran subroutine). In this case one must supply the mapping from the unit cube to the computational domain and the derivatives of this mapping. This approach is useful if a grid has been created by some other package. If the component grid mapping is only known at a set of grid points then the mapping can be defined everywhere by using interpolation. CMPGRD has such a component grid interpolation routine.

Another way to define a three-dimensional grid is to first define a surface in three-dimensions as an externally defined curve. This surface can then be automatically extended in the normal direction to create a three-dimensional grid. Alternatively two offset surfaces can be blended to form a space filling grid.

Sphere in a Box: As a first example of creating a three-dimensional composite grid we consider the problem of generating a smooth grid for the region exterior to a sphere and interior to a box in which the sphere sits. We describe two ways to create a composite grid for this problem.

A grid around the sphere can be created using a mapping defined by the standard spherical-polar coordinate transformation:

$$\begin{aligned} \text{Sphere} : (r_1, r_2, r_3) &\rightarrow (x_1, x_2, x_3) \\ (\theta, \phi, R) &= (2\pi r_1, \pi(r_2 - .5), R_a + (R_b - R_a) r_3) \\ (x_1, x_2, x_3) &= (R\cos(\theta)\cos(\phi), R\sin(\theta)\cos(\phi), R\sin(\phi)) \end{aligned}$$

This is not the recommended method to create a grid since the mapping has singularities at the north and south poles. These singularities will spell trouble when one wants to solve a PDE on the grid. In fact CMPGRD must know which grids are singular in order for its algorithm to work properly.

By using multiple patches it is possible to cover the surface of a sphere with grid transformations so that there are no coordinate singularities. At each pole we create a grid defined by an orthographic projection, see figure (1). Consider the set of points $(s_1, s_2, -R)$ on the plane which is tangent to the sphere (radius R) at the south pole. The orthographic projection associates a point $(s_1, s_2, -R)$ on this plane with a point $\mathbf{x} = (x_1, x_2, x_3)$ on the sphere as the intersection between the sphere and the line through the north pole and the point $(s_1, s_2, -R)$. This point of intersection is given by

$$\begin{aligned} \text{Orthographic} : (s_1, s_2) &\rightarrow (x_1, x_2, x_3) \\ (x_1, x_2, x_3) &= (\rho\cos(\theta), \rho\sin(\theta), -\zeta) \\ &= (\frac{\alpha^2 s_1}{(\alpha^2 + s^2)}, \frac{\alpha^2 s_2}{(\alpha^2 + s^2)}, \alpha\frac{(\alpha^2 - s^2)}{(\alpha^2 + s^2)}) \end{aligned}$$

where

$$\alpha = 2R \quad , \quad s^2 = s_1^2 + s_2^2 \quad , \quad \cos(\theta) = \frac{s_1}{s} \quad , \quad \sin(\theta) = \frac{s_2}{s}$$

$$\rho = \alpha\frac{\alpha^2 s}{\alpha^2 + s^2} \quad , \quad \zeta = \alpha\frac{\alpha^2 - s^2}{\alpha^2 + s^2} \quad , \quad \zeta^2 + \rho^2 = R^2$$

To complete the definition of the grid transformation we define the relationship between the unit square coordinates (r_1, r_2) and the (s_1, s_2) coordinates,

$$s_1 = (r_1 - .5)\ s_a \quad , \quad s_2 = (r_2 - .5)\ s_b$$

where the constants s_a, s_b determine the extent of the orthographic patch. For later reference we note the derivatives of the orthographic transformation:

$$\frac{\partial\rho}{\partial s}=\frac{\rho\,\zeta}{s\,R}\quad,\quad\frac{\partial\zeta}{\partial s}=-\frac{\rho^2}{R}\quad,\quad\frac{d\zeta}{d\rho}=\frac{-\rho}{\zeta} \tag{1}$$

$$\frac{\partial s}{\partial s_1}=\frac{s_1}{s}\quad,\quad\frac{\partial s}{\partial s_2}=\frac{s_2}{s}\quad,\quad\frac{\partial\theta}{\partial s_1}=-\frac{\sin(\theta)}{s}\quad,\quad\frac{\partial\theta}{\partial s_2}=\frac{\cos(\theta)}{s} \tag{2}$$

The full three-dimensional component grid is defined by extending each surface in the normal direction. The box is covered with a simple rectangular grid. The composite grid is shown in figures (3) and (4).

Surfaces defined by cross-sections: The method described in the previous section can be generalized to create grids for surfaces defined by cross sections. We assume that the surface is defined as a function of a periodic variable θ, $0\le\theta\le 2\pi$, and an axial variable ζ, $-1\le\zeta\le +1$:

$$\mathbf{x}=\mathbf{f}(\theta,\zeta)$$

When ζ is fixed $\mathbf{f}(\theta,\zeta=\zeta_0)$ will define a cross-sectional curve on the surface. The cross-sections should converge to a point at $\zeta=\pm1$. We will also see that the cross-sections must also tend to an elliptical shape as $\zeta\rightarrow\pm1$ in order that there be no singularity in the derivative of the mapping at the poles.

The cross-sectional surface will be covered by three patches: a central cylindrical patch and two end patches defined by orthographic projections. The central patch is defined by the mapping

$$\begin{aligned}C_2:(r_1,r_2)&\;\longrightarrow\;(x_1,x_2,x_3)\\ \mathbf{x}&\;=\;\mathbf{f}(\theta,\zeta)\\ (\frac{\partial\mathbf{x}}{\partial r_1},\frac{\partial\mathbf{x}}{\partial r_2})&\;=\;(2\pi\,\frac{\partial\mathbf{f}}{\partial\theta},\zeta_c\,\frac{\partial\mathbf{f}}{\partial\zeta})\\ \text{where }\;\theta=2\pi\;r_1\quad&,\quad\zeta=\zeta_c\,(r_2-.5)\end{aligned}$$

The constant $\zeta_c\le 2$ determines the portion of the sphere covered by the central patch. When $\zeta_c=2$ the central patch will extend all the way to each pole. The orthographic patch at the $\zeta=+1$ pole is defined by

$$C_1:(r_1,r_2)\;\longrightarrow\;(x_1,x_2,x_3) \tag{3}$$

$$\mathbf{x}\;=\;\mathbf{f}(\theta,\zeta) \tag{4}$$

$$\frac{\partial\mathbf{x}}{\partial r_1}\;=\;\frac{\rho}{s}\Big[-\sin(\theta)\{\frac{1}{\rho}\frac{\partial\mathbf{f}}{\partial\theta}\}-\frac{\cos(\theta)}{R}\{\rho\frac{\partial\mathbf{f}}{\partial\zeta}\}\Big] \tag{5}$$

$$\frac{\partial\mathbf{x}}{\partial r_2}\;=\;\frac{\rho}{s}\Big[\;\cos(\theta)\{\frac{1}{\rho}\frac{\partial\mathbf{f}}{\partial\theta}\}-\frac{\sin(\theta)}{R}\{\rho\frac{\partial\mathbf{f}}{\partial\zeta}\}\Big] \tag{6}$$

where ρ, s and θ are the variables associated with the orthographic patch:

$$s_1=(r_1-.5)\;s_a\quad,\quad s_2=(r_2-.5)\;s_b\quad,\quad s^2=s_1^2+s_2^2$$

$$\cos(\theta)=\frac{s_1}{s}\quad,\quad\sin(\theta)=\frac{s_2}{s}\quad,\quad\rho=\frac{\alpha s}{\alpha^2+s^2}\quad,\quad\zeta=\alpha\frac{\alpha^2-s^2}{\alpha^2+s^2}\quad,\quad\rho^2+\zeta^2=R^2$$

The Jacobian derivatives of the mapping, equations (5), (6) are determined in the standard way using the chain rule

$$\begin{aligned}\frac{\partial \mathbf{x}}{\partial r_1} &= \frac{\partial \theta}{\partial r_1}\frac{\partial \mathbf{f}}{\partial \theta} + \frac{\partial \zeta}{\partial r_1}\frac{\partial \mathbf{f}}{\partial \zeta} \\ &= \frac{-\sin(\theta)}{s}\frac{\partial \mathbf{f}}{\partial \theta} + \frac{-\rho^2}{R}\cos(\theta)\frac{\partial \mathbf{f}}{\partial \zeta} \ .\end{aligned}$$

These derivatives are written in the special form of equations (5),(6). We assume that we have a subroutine which defines the cross-sections $\mathbf{f}(\theta,\zeta)$ and also certain derivatives of the cross-sections:

$$\text{Cross-section routine: } (\theta,\zeta) \ \longrightarrow \ (\mathbf{f}(\theta,\zeta), \frac{1}{\rho}\frac{\partial \mathbf{f}}{\partial \theta}, \rho\frac{\partial \mathbf{f}}{\partial \zeta}) \tag{7}$$

We choose to have the routine return $(\frac{1}{\rho}\frac{\partial \mathbf{f}}{\partial \theta}, \rho\frac{\partial \mathbf{f}}{\partial \zeta})$ as opposed to $(\frac{\partial \mathbf{f}}{\partial \theta}, \frac{\partial \mathbf{f}}{\partial \zeta})$ in order to avoid inaccuracy in numerically computing the Jacobian. In fact, in order that the derivatives of the patches exist at the poles the cross-sectional function $\mathbf{f}$ must satisfy

$$\lim_{s\to 0} \ \Big[\cos(\theta)\frac{\partial \mathbf{f}}{\partial \rho} - \frac{\sin(\theta)}{s}\frac{\partial \mathbf{f}}{\partial \theta}\Big] = C_1 \tag{8}$$

$$\lim_{s\to 0} \ \Big[\sin(\theta)\frac{\partial \mathbf{f}}{\partial \rho} + \frac{\cos(\theta)}{s}\frac{\partial \mathbf{f}}{\partial \theta}\Big] = C_2 \tag{9}$$

where the constants C_1, C_2 must be independent of θ. These conditions will be satisfied provided that the cross-sections tend to an elliptical shape at the poles:

$$f_i(\theta,\zeta) \ \sim \ \rho(a\cos(\theta) + b\sin(\theta)) \ .$$

Condition (8), for example, follows from writing equations (5) in the form

$$\frac{\partial \mathbf{x}}{\partial r_1} \ = \ \Big[\cos(\theta)\frac{\partial \mathbf{f}}{\partial \rho} - \frac{\sin(\theta)}{s}\frac{\partial \mathbf{f}}{\partial \theta}\Big] + \frac{1}{s^2}\Big(\frac{\rho}{s}\frac{\zeta}{R} - 1\Big)(s\cos(\theta))\Big(s\frac{\partial \mathbf{f}}{\partial \rho}\Big)$$

and noting that

$$\lim_{\rho\to 0}\frac{1}{s^2}\Big(\frac{\rho}{s}\frac{\zeta}{R} - 1\Big)(s\cos(\theta)) = 0 \ .$$

Ellipsoid: A grid for an ellipsoid with principal axes of lengths (a,b,c) can be created using the cross-section function

$$\begin{aligned}\text{Ellipsoid} : (\theta,\zeta) \ &\longrightarrow \ (x_1,x_2,x_3) \\ (x_1,x_2,x_3) \ &= \ (f_1,f_2,f_3) = (a\rho\cos(\theta), b\rho\sin(\theta), c\zeta)\end{aligned}$$

with Jacobian derivatives defined by

$$\begin{bmatrix} \frac{1}{\rho}\partial f_1/\partial\theta & \frac{1}{\rho}\partial f_2/\partial\theta & \frac{1}{\rho}\partial f_3/\partial\theta \\ \rho\partial f_1/\partial\zeta & \rho\partial f_2/\partial\zeta & \rho\partial f_3/\partial\zeta \end{bmatrix} = \begin{bmatrix} -a\sin(\theta) & b\cos(\theta) & 0 \\ -a\zeta\cos(\theta) & -b\zeta\sin(\theta) & c\rho \end{bmatrix}$$

The resulting grid is shown in figure (5).

Wing with Joukowsky Cross-Section: A slightly more complicated problem involves the generation of a grid around an wing which sits in a box. For simplicity we define the wing to have cross-sections which are Joukowsky airfoils. At the wing tip the cross-sections are forced to become elliptical since this is a necessary requirement for the grid to be smooth. The mappings are thus defined as

$$\begin{aligned} \text{Joukowsky wing} : (r_1, r_2) &\rightarrow (x_1, x_2, x_3) \\ (x_1, x_2, x_3) &= (f_1(\zeta), g(\rho) f_2(\theta, \zeta), g(\rho) f_3(\theta, \zeta)) \end{aligned}$$

The cross section functions f_2 and f_3 are defined as Joukowsky airfoils:

$$\begin{aligned} f_2 + i f_3 &= w + 1/w \\ w &= a e^{i\theta} + i d e^{i\delta} f_4(\rho) \\ f_4 &= g(\rho) - g'(1)\rho \end{aligned}$$

The function $g(\rho)$ (recall $\rho = \sqrt{1 - \zeta^2}$) is chosen to be $g(\rho) = \tanh(\beta\rho)$ and causes the cross-sections to converge to a point. The function f_4 satisfies $f_4(\rho = 0) = 0$ causing the airfoil to become elliptical at the tip and $f_4'(\rho = 1) = 0$ causing the cross-section at $(\zeta, \rho) = (0, 1)$ to lie in the in the $x_2 - x_3$ plane. We make use of this latter result when we attach the wing to a fuselage in the next section. For the airfoil shown the free parameters defining the airfoil were taken as

$$a = .85 \quad , \quad d = .15 \quad , \quad \delta = 15^\circ = 15(\pi/180)^r \quad , \quad \beta = 2$$

In order to have sufficient grid lines in locations of highest curvature we stretch the grid lines at the leading and trailing edges.

Wing on a fuselage - first approach: In this section we consider the problem of attaching a wing to a fuselage. Here we describe a relatively simple procedure to accomplish the joining. Later we describe another solution to this problem. The procedure is illustrated in figure (7) and consists of the steps:

1. Create component grids for the wing and for the fuselage.
2. Deform the end of the wing grid to be the same shape as the piece of the fuselage it will attach to.
3. Position the deformed wing onto the fuselage.

For step 1 we use the wing grid as defined in the previous section. For simplicity we define the fuselage to be part of a cylinder:

$$\begin{aligned} \text{Cylinder} : (r_1, r_2) &\rightarrow (x_1, x_2) \\ \theta = \pi(r_1 - .5) \quad &, \quad R = R_1 + R_2 r_2 \\ (x_1, x_2, x_3) &= (R\cos(\theta), y_a + y_{ba} r_3, R\sin(\theta)) \ . \end{aligned}$$

To perform step 2 we deform the end of the wing to lie on a cylinder. For our geometry this is accomplished with the mapping

$$x_1 \leftarrow x_1 + R_1 - \left\{ R_1 - \sqrt{R_1^2 - x_3^2}\, e^{-\alpha x_1} \right\}$$

which transforms the end of the airfoil onto the cylinder. The mapping has an exponentially small effect on parts of the wing which are far from the end (figure (7b). Finally we move the wing to join the fuselage. The resulting composite grid will have the topology shown in figure (7c).

In general, to perform step 2 of deforming the end of the wing we will need to know how to map a plane (or the end face of the wing grid) onto the surface of the fuselage. It is also important to perform this deformation without changing the surface shape of the wing. The composite grid for a Joukowsky wing attached to a cylinder in a box is shown in figure (6). The figure shows a cut through the wing and cylinder. The interpolation points are shown.

This example illustrates some important advantages and disadvantages of composite grids. The advantage was gained because the grid generation problem was simplified by decomposing the domain into simpler problems. The disadvantage came when the grids had to be joined together along a physical boundary.

M6 wing: In this section we use the ideas of the previous section to create a grid for the ONERA M6 wing which is free from artificial coordinate singularities. The M6 wing does, however, have a sharp trailing edge. The geometry of the M6 wing is defined by a sequence of cross-sections. Each cross-section is defined by a set of points. We fit a cubic spline to each cross section, parameterized by pseudo-arclength. Thus if the cross-section at $\zeta = \zeta_i$ is defined by the points

$$(f_2(\theta,\zeta_i), f_3(\theta,\zeta_i)) \ : \ \{(y_j, z_j)\}_{j=1}^{n_j}$$

then we define the pseudo-arclength by $s_1 = 0$ and

$$s_j = s_{j-1} + C\sqrt{(y_j - y_{j-1})^2 + (z_j - z_{j-1})^2} \quad j = 2, 3, \ldots, n_j \ ,$$

where C is chosen so that $s_{n_j} = 1$. We fit cubic splines

$$\begin{aligned} Y_i(s) &= \text{spline}(\{(s_j, y_j)\}) \\ Z_i(s) &= \text{spline}(\{(s_j, z_j)\}) \end{aligned}$$

We use (linear) interpolation between cross-sections to define the surface of the wing everywhere:

$$\begin{aligned} f_2(\theta,\zeta) &= (1-\alpha)Y_i(s) + \alpha Y_{i+1}(s) \\ f_3(\theta,\zeta) &= (1-\alpha)Z_i(s) + \alpha Z_{i+1}(s) \\ s &= \theta/(2\pi) \\ \alpha &= (\zeta - \zeta_i)/(\zeta_{i+1} - \zeta_i) \ , \quad \zeta_i \le \zeta \le \zeta_{i+1} \end{aligned}$$

The cross-sections at the wing tip should converge to a point. Moreover in order that the tip be smooth the cross-sections should converge to an ellipse. To ensure these conditions we slightly deform the tip of the wing. The definition of the cross-sections for the wing are thus

$$\begin{aligned} \text{M6 wing} : (r_1, r_2) &\rightarrow (x_1, x_2, x_3) \\ (x_1, x_2, x_3) &= (L\zeta, \ \tanh(\beta\rho) f_2(\theta,\zeta), \ \tanh(\beta\rho) f_3(\theta,\zeta)) \end{aligned}$$

where L is the length of the wing and where β is chosen sufficiently large so as to not significantly change the shape of the wing tip. The composite grid for an M6 attached to a cylinder is shown in figure (8).

Component grids created from CAD Package: A common representation of surfaces in computer aided design (CAD) packages is that of a Coons patch [8], also known as transfinite interpolation [9]. The Coons patch is defined as a mapping from the unit square into three-space. In this representation the unit square (r_1, r_2) is divided into an array of sub-patches. There are $n_i \times n_j$ such subpatches. Each subpatch is defined as a polynomial, mapping the unit square (u_1, u_2) (of the subpatch) into $\mathbf{x} \in R^3$.

$$\text{Subpatch: } \mathbf{x}_{ij}(u_1, u_2) = \sum_{m=0}^{n_u-1} \sum_{n=1}^{n_v-1} \hat{\mathbf{x}}^{ij}_{mn} u_1^m u_2^n \tag{10}$$

where $\hat{\mathbf{x}}^{ij}_{mn}$ are constants. The subpatch boundaries, $r_1 = r_{1,i}$, $i = 1, ..., n_i - 1$ and $r_2 = r_{2,j}$, $j = 1, ..., n_j - 1$, are equally spaced in the (r_1, r_2) plane,

$$r_{1,i} = \frac{i}{n_1 + 1} \quad , \quad i = 1, ..., n_i \quad , \quad r_{2,i} = \frac{j}{n_2 + 1} \quad , \quad j = 1, ..., n_j \ .$$

The full mapping is defined as

$$\mathbf{x}(r_1, r_2) \quad = \quad \mathbf{x}_{ij}\left(\frac{r_1 - r_{1,i}}{r_{1,i+1} - r_{1,i}}, \frac{r_2 - r_{2,j}}{r_{2,j+1} - r_{2,j}}\right) \quad \text{for} \quad r_{1,i} \leq r_1 \leq r_{1,i+1} \quad \text{and} \quad r_{2,j} \leq r_2 \leq r_{2,j+1}$$

The smoothness of the multi-patch surface $\mathbf{x}(r_1, r_2)$ is obtained by placing constraints on the sub-patch mappings at the boundaries of the sub-patches. Often patches are not parametrically smooth but only geometrically smooth: the derivatives with respect to r_1 or r_2 may not be continuous across different subpatches. In general we require parametric smoothness of surfaces if we wish to use the grid to solve a PDE problem. In this case we may want to reparameterize the patch to create a smooth parameterization. We are working on a variety of methods for this purpose. In figure (9) we show a composite grid for some surfaces created from the CATIA computer aided design package. This grid shows a wing connected to an engine nacelle by a pylon.

CREATING COMPONENT GRIDS FROM INTERSECTING SURFACES

Curves of Intersection: The first step in defining a grid in the region where two surfaces intersect is to define and parameterize the curve of intersection. Even assuming that the curve of intersection is well defined, consisting of one connected component, the parameterization of this curve is not well defined. This parameterization will be important when we wish to create a grid.

Suppose we have two surfaces, $\mathbf{S}_i : [0,1]^2 \rightarrow \mathbf{R}^3$, $i = 1, 2$ mapping the unit square into three-dimensional space,

$$\mathbf{S}_i \quad = \quad \{\mathbf{x} = \mathbf{S}_i(\mathbf{r}_i) \ : \ \mathbf{r}_i = (r_i, s_i) \in [0,1]^2\}$$

and suppose these surfaces intersect along some curve (figure (10)). Denote this curve of intersection by $\mathbf{C} : [0,1] \rightarrow \mathbf{R}^3$,

$$\mathbf{C} = \{\mathbf{x} = \mathbf{C}(s) \ : \ s \in [0,1]\}.$$

As the curve $\mathbf{C}(s)$ traces out a path on the surface $\mathbf{S}_i$ it will also trace out a path on the unit square which is mapped to the surface $\mathbf{S}_i$. Denote this two-dimensional curve on the unit square of $\mathbf{S}_i$ by $\mathbf{R}_i : [0,1] \rightarrow [0,1]^2$,

$$\mathbf{R}_i \quad = \quad \{\mathbf{r} = \mathbf{R}_i(s) \ : \ s \in [0,1]\}.$$

The curves $\mathbf{R}_i(s)$ satisfy

$$\mathbf{C}(s) \;=\; \mathbf{S}_i(\mathbf{R}_i(s)) \qquad s \in [0,1], \;\; i = 1,2 \tag{11}$$

We see that the curve of intersection can be represented in three ways, $\mathbf{C}(s)$, $\mathbf{S}_1(\mathbf{R}_1(s))$ and $\mathbf{S}_2(\mathbf{R}_2(s))$. For consistency we want a numerical representation for these curves so that the defining conditions (11) are satisfied precisely (up to round off errors). This means that when we evaluate the curve we must solve the nonlinear equations (11), using Newton's method for example. That is given a value of s we determine $\{\mathbf{C}(s), \mathbf{R}_1(s), \mathbf{R}_2(s)\}$ satisfying (11). We will keep an accurate approximation to the curve to use as a starting guess for Newton. One reason we require such a precise definition for the curve of intersection is that the curve may be used to define a grid which has very small spacings between grid lines. In this case a less accurate representation of $\mathbf{C}$ or $\mathbf{R}_i$ may lead to erroneous results when interpolation points are computed for the overlapping grid.

One way to choose the parameterization of $\mathbf{C}(s)$ is to use one of r_1, s_1, r_2, s_2 to parameterize the curve. For example if the curve $\mathbf{R}_1(s)$ is a single valued function of s_1, as in figure (10), then we may use $s = r_1$. The equations defining $\mathbf{C}(s)$ would be

$$\begin{aligned} \mathbf{C}(s) &= \mathbf{S}_i(\mathbf{r}_i) \quad i = 1,2 \\ s_1 &= s \end{aligned} \tag{12}$$

For given s there are 7 equations for the 7 unknowns $\mathbf{C}(s)$, $\mathbf{r}_i = \mathbf{R}_i(s)$, $i = 1,2$. These equations can be solved by bisection in the variable r_1. Another way to choose the parameterization of $\mathbf{C}(s)$ is to base the parameterization on another curve which is close to $\mathbf{C}$. Thus suppose we have another approximate intersection curve $\mathbf{C}_A(s)$. Given s we define $\mathbf{C}(s)$ to be the point on $\mathbf{C}$ which intersects the plane which is normal to the curve $\mathbf{C}_A$ at the point $\mathbf{C}_A(s)$, see figure (11).

$$\begin{aligned} \mathbf{C}(s) &= \mathbf{S}_i(\mathbf{r}_i) \quad i = 1,2 \\ \frac{\partial \mathbf{C}_A}{\partial s} \cdot (\mathbf{C}(s) - \mathbf{C}_A(s)) &= 0 \end{aligned}$$

These seven equations define the seven unknowns $\mathbf{C}(s)$, $\mathbf{r}_i = \mathbf{R}_i(s)$, $i = 1,2$.

Component grids defined using the curves of intersection: We consider a few examples showing how a component grid can be constructed making use of the curves of intersection. In these examples the curve of intersection, $\mathbf{C}(s)$, between the two surfaces will become the edge of a component grid. The faces of the component grid adjacent to this edge will lie on the respective surfaces. To parameterize these faces we will make use of the curves $\mathbf{R}_i(s)$, corresponding to $\mathbf{C}(s)$, which lie in the unit square coordinates. By construction, the grid we create will have faces that lie precisely (up to round off errors) on the surfaces.

The geometry of the first example is shown in figure (12) where two concentric cylinders intersect two planes. The goal is to create the grid $\mathbf{x} = \mathbf{G}(\mathbf{r})$ which has faces coming from the cylinders and planes as shown in figure (13). Figure (12) shows the 4 unit squares and the curves of intersection. To define the faces of $\mathbf{G}$ we use the portions of the surfaces which are bounded by the curves of intersection. In the unit square coordinates the faces will correspond to the regions bounded by the images of the curves of intersection. The mapping for the face, $\mathbf{G}_1(r_1, r_2, 0)$ corresponding to the end at $r_3 = 0$ of $\mathbf{G}$ can be defined as the composite transformation,

$$\mathbf{G}_1(r_1, r_2, 0) = \mathbf{S}_1(\mathbf{P}_1(r_1, r_2)),$$

where the mapping $\mathbf{P}_1$ in the $\mathbf{r}_1$ plane can be defined, for example, by transfinite interpolation

$$\mathbf{r} = \mathbf{P}_1(r_1, r_2) = (1 - r_2)\mathbf{R}_1(r_1) + r_2\ \mathbf{R}_2(r_1).$$

The other three faces are defined in a similar manner (figure 13). The grid $\mathbf{G}$ is then defined as a transfinite interpolation between these four faces

$$\begin{aligned}\mathbf{G}(r_1, r_2, r_3) \;=\;& (1 - r_1)\mathbf{G}_1(0, r_2, r_3) + r_1\mathbf{G}(1, r_2, r_3) + (1 - r_3)\mathbf{G}_1(r_1, r_2, 0) + r_3\mathbf{G}(r_1, r_2, 1)\\ & -\Big\{(1 - r_1)\Big[(1 - r_3)\mathbf{G}_1(0, r_2, 0) + r_3\mathbf{G}(0, r_2, 1)\Big]\\ & \quad +r_1\Big[(1 - r_3)\mathbf{G}_1(1, r_2, 0) + r_3\mathbf{G}(1, r_2, 1)\Big]\Big\}\end{aligned}$$

Figure (14) shows some surfaces and a grid that have been created using these techniques. Note that this parameterization of the grid $\mathbf{G}$ will only be smooth if the parameterizations of the curves of intersection, $\mathbf{C}_i(s)$ are defined in a consistent way. A major catastrophe occurs, for example, if opposite faces have parameterizations defined in opposite directions so the grid folds over on itself. In a less severe case the grid lines may become highly skewed. Even if the parameterization is still single valued we wish to avoid such cases since finite difference approximations are usually less accurate for skewed grids.

Grid for two intersecting spheres: In the second example we create a grid to be used in the region where two spheres (actually spherical shells) intersect. This *collar* grid will be used to join the spheres and looks something like a torus. Two of the faces of this collar grid lie on the the surfaces of the spheres. The other two faces lie on spherical shells which are offset from the spheres. The collar grid is shown in figure (15). The composite grid for the two spheres in shown in figure (16). Only the grids on the surfaces of the spheres are shown but the collar grid can be recognized.

Wing on a fuselage - approach 2: We use the techniques for creating grids from intersecting surfaces to connect a wing to a fuselage. We first define surfaces for a cylinder and a wing with Joukowsky cross-sections. Another wing-like surface is defined by offsetting the wing in the normal direction. The two wing surfaces are intersected with the cylindrical surface and a wing grid is defined in the region bounded by the intersections. The composite grid is shown in figure (17).

CONCLUSIONS

We have given a brief description of some techniques for the construction of overlapping grids in three space dimensions. We have shown how to create grids for surfaces that are defined by cross-sections. We have described some ways to create grids in the regions where surfaces intersect. These grids are defined by smooth transformations and are free from artificial coordinate singularities. The ability to create smooth grids for complicated regions is an important first step towards the accurate numerical solution of PDEs on such regions. The results presented in this paper show some of the potential advantages of using overlapping grids to create grids for three dimensional geometries. However, there is still much work to be done to make the grid construction problem easier and more automatic.

References

[1] G. Chesshire and W. D. Henshaw. Composite overlapping meshes for the solution of partial differential equations. *J. Comp. Phys.*, 90(1):1–64, 1990.

[2] W. D. Henshaw and G. Chesshire. Multigrid on composite meshes. *SIAM J. Sci. Stat. Comput.*, 8(6):914–923, 1987.

[3] P. G. Buning, I. T. Chiu, S. Obayashi, Y. M. Rizk, and J. L. Steger. Numerical simulation of the integrated space shuttle vehicle in ascent. paper 88-4359-CP, AIAA, 1988.

[4] J. L. Steger and J. A. Benek. On the use of composite grid schemes in computational aerodynamics. *Computer Methods in Applied Mechanics and Engineering*, 64:301–320, 1987.

[5] D. L. Brown. A finite volume method for solving the Navier-Stokes equations on composite overlapping grids. In B. Engquist and B. Gustafsson, editors, *Third International Conference on Hyperbolic Problems*, pages 141–158. Chartwell-Bratt, 1991.

[6] G. Chesshire and W. D. Henshaw. Conservation on composite overlapping grids. IBM Research Report RC 16531, IBM Research Divison, Yorktown Heights, NY, 1991.

[7] D. L. Brown, G. Chesshire, and W. D. Henshaw. Getting started with CMPGRD, introductory user's guide and reference manual. report LA-UR-89-1294, Los Alamos National Laboratory, 1989.

[8] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design.* Academic Press, 1988.

[9] J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin. *Numerical Grid Generation.* North-Holland, New York, 1985.

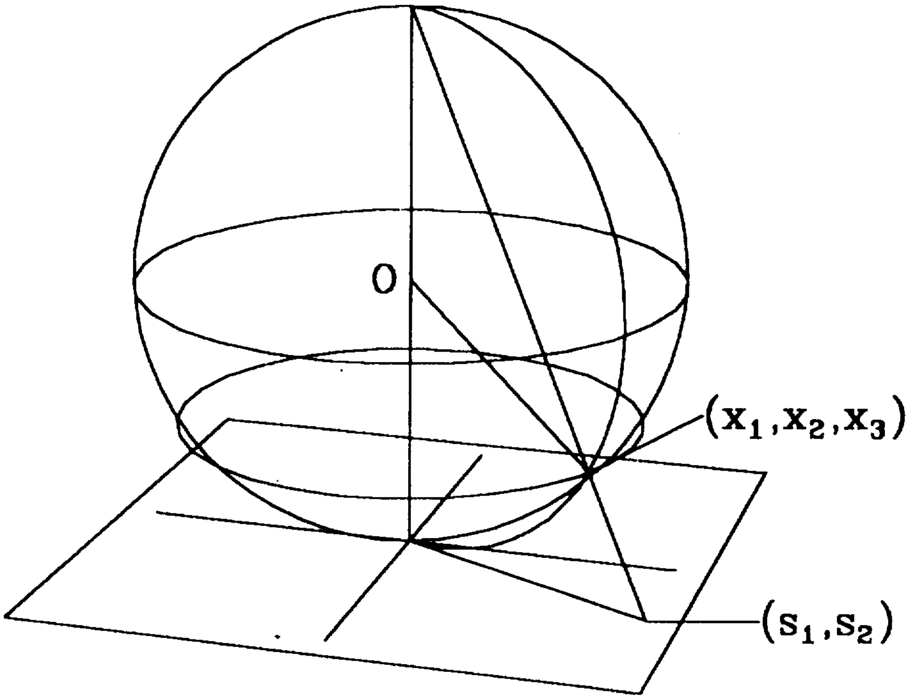

Figure 1: Orthographic projection

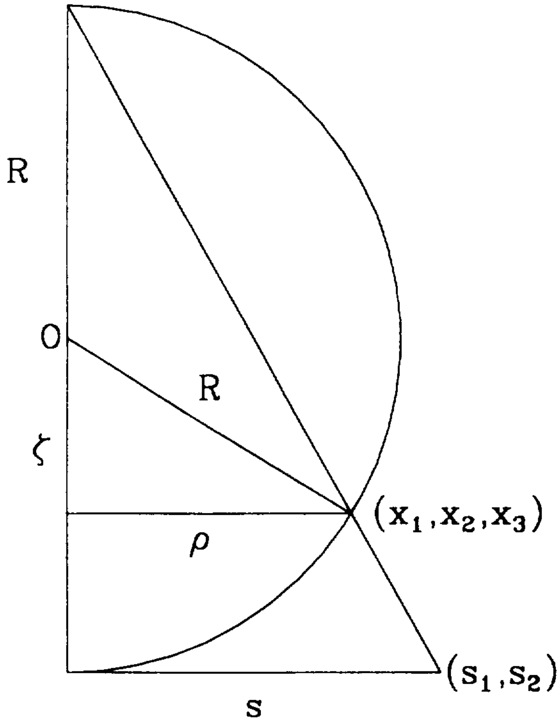

Figure 2: Variables associated with the orthographic projection

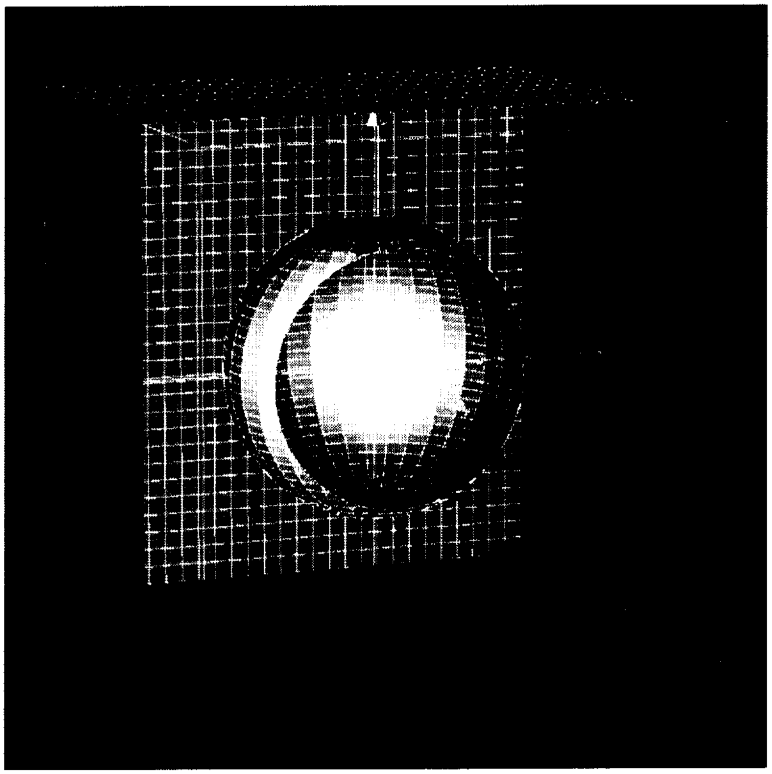

Figure 3: Overlapping grid for a sphere

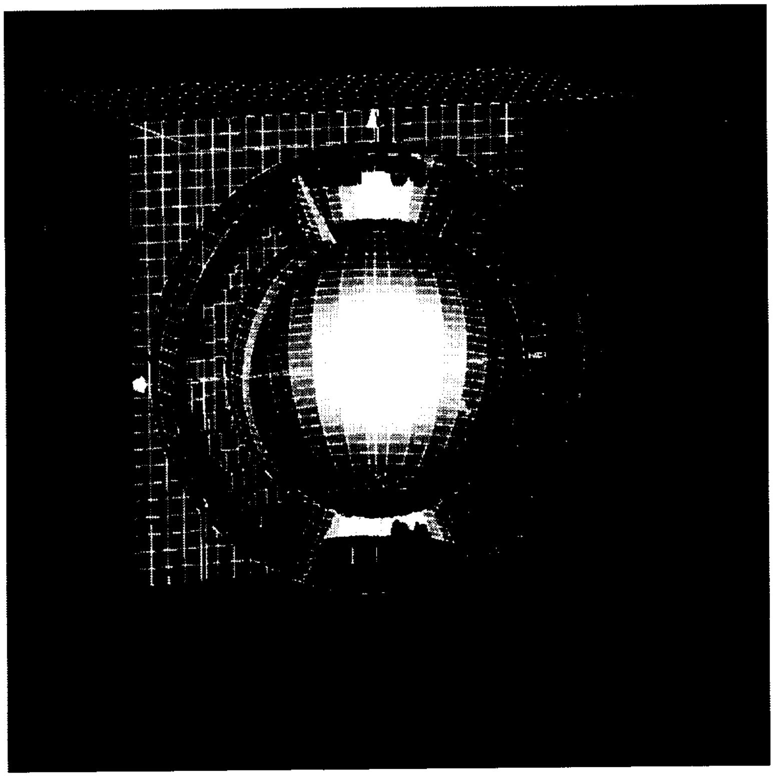

Figure 4: Overlapping grid for a sphere showing interpolation points

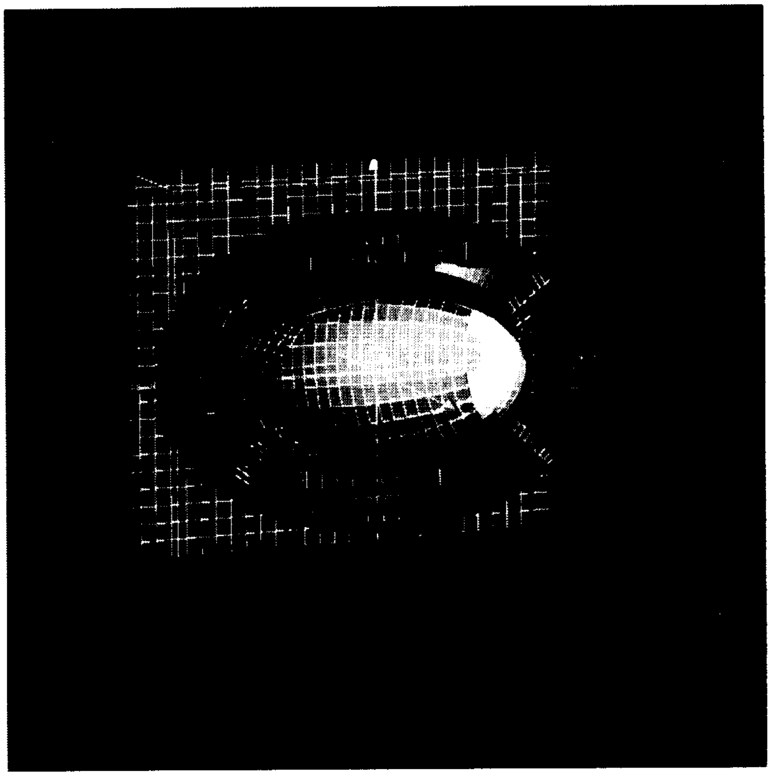

Figure 5: Overlapping grid for an ellipse showing interpolation points

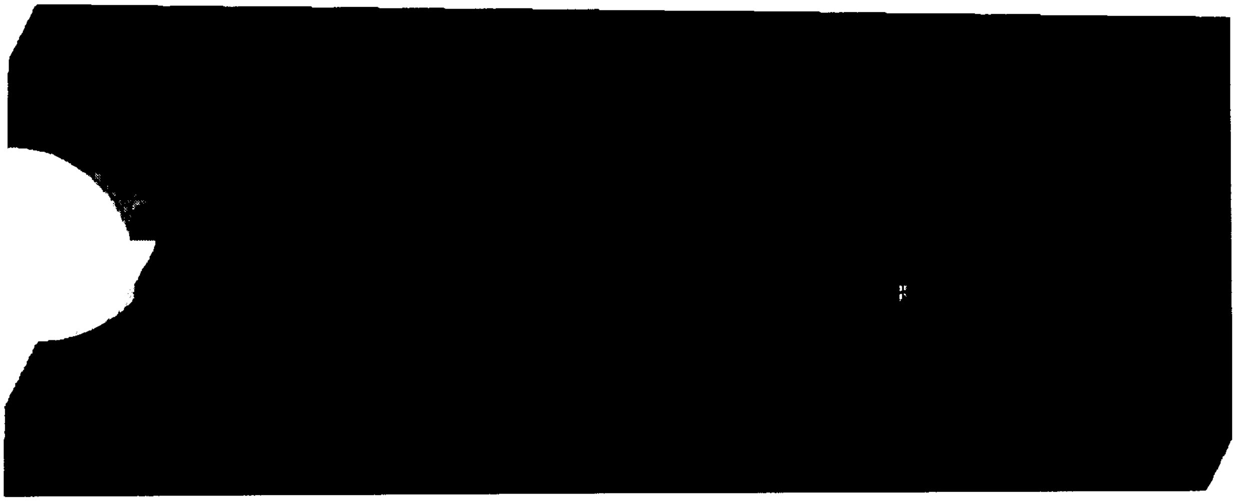

Figure 6: Overlapping grid for Joukowsky wing on a cylinder

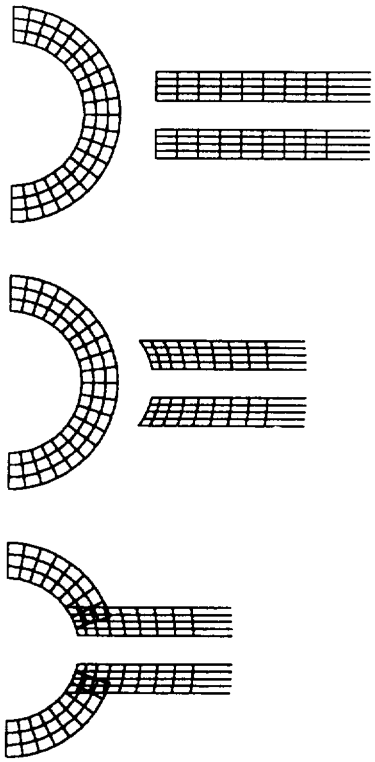

Figure 7: Attaching a wing to a fuselage - approach I

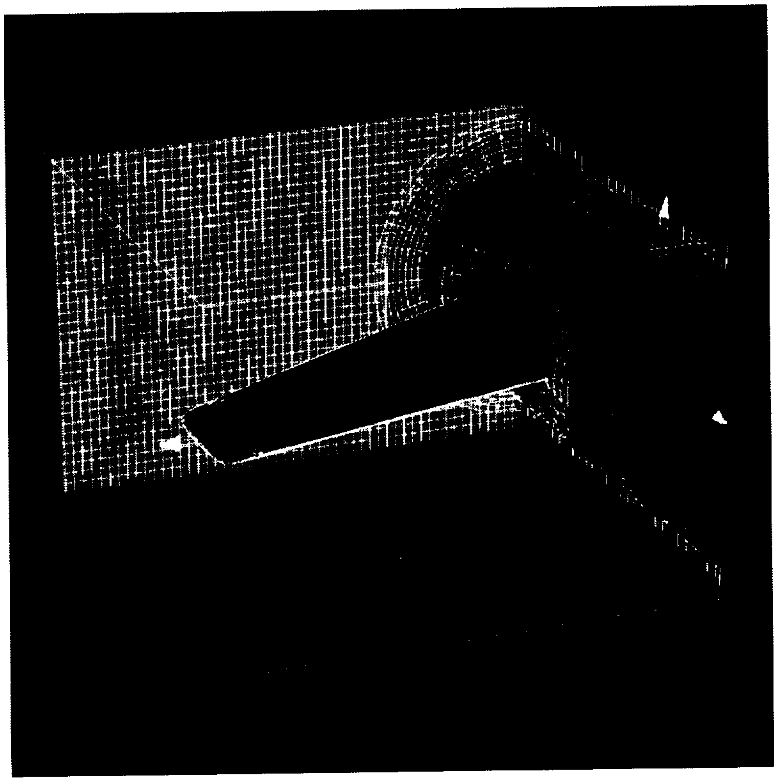

Figure 8: Overlapping grid for an M6 wing on a cylinder

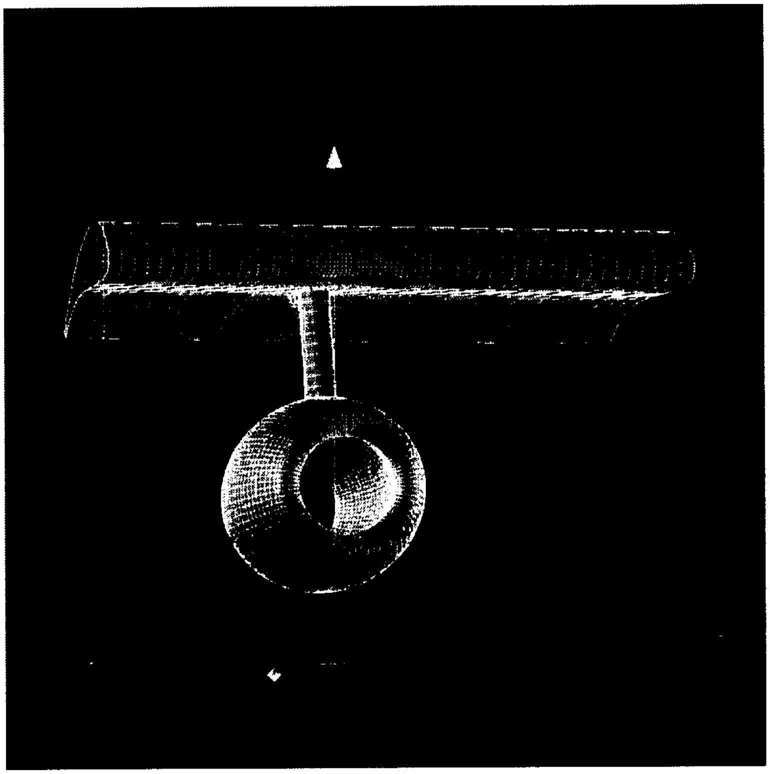

Figure 9: Overlapping grid for a wing, pylon and engine nacelle

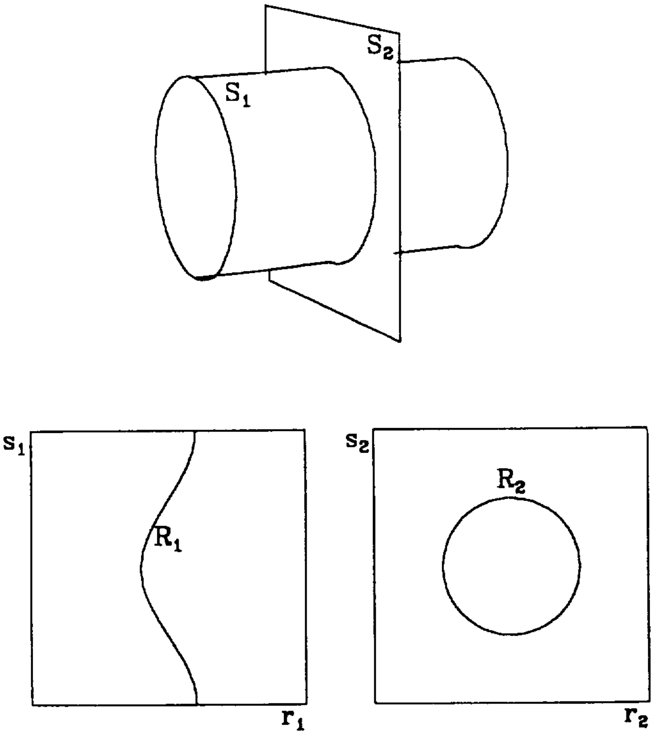

Figure 10: Curve of intersection for a cylinder and a plane

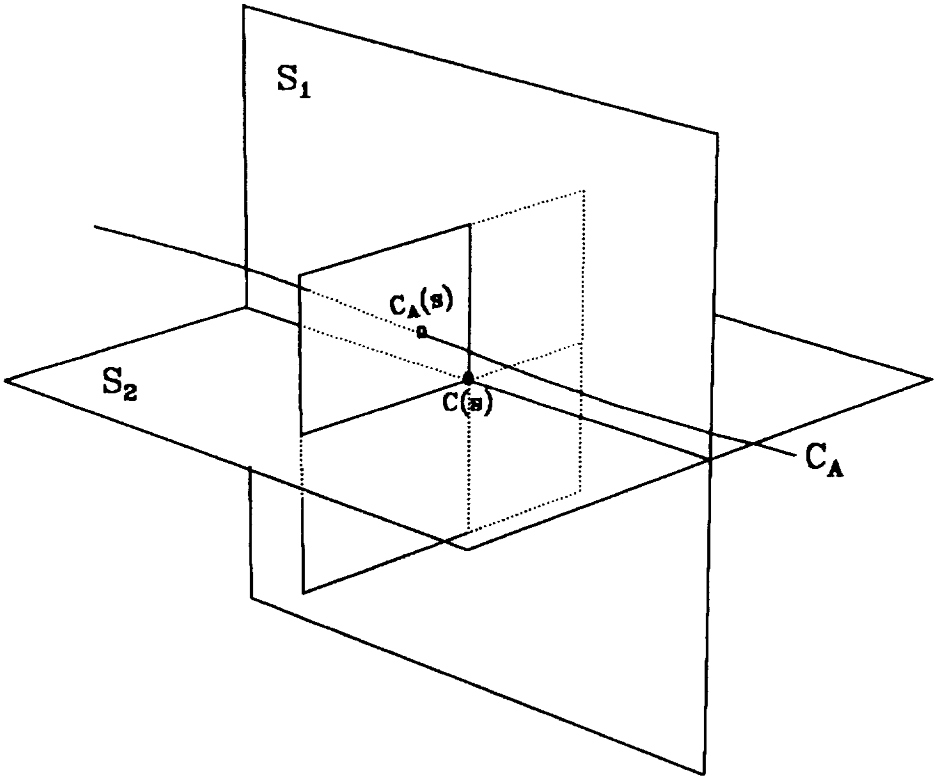

Figure 11: Parameterizing the curve of intersection with a nearby curve

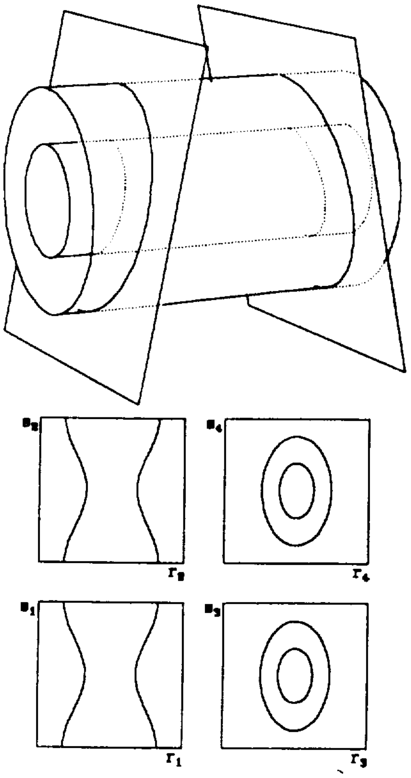

Figure 12: Curves of intersection for G_1

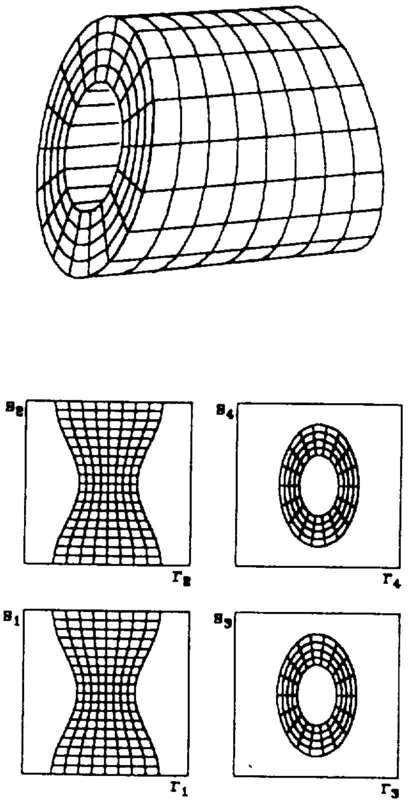

Figure 13: Grid G_1 and the images of the faces in the unit squares

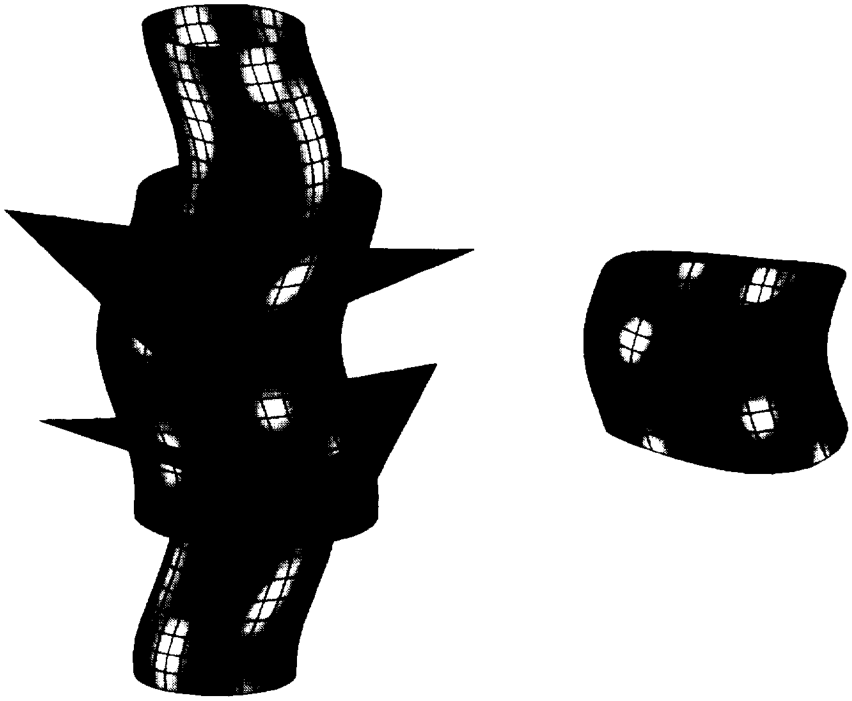

Figure 14: Creating a component grid by intersecting surfaces

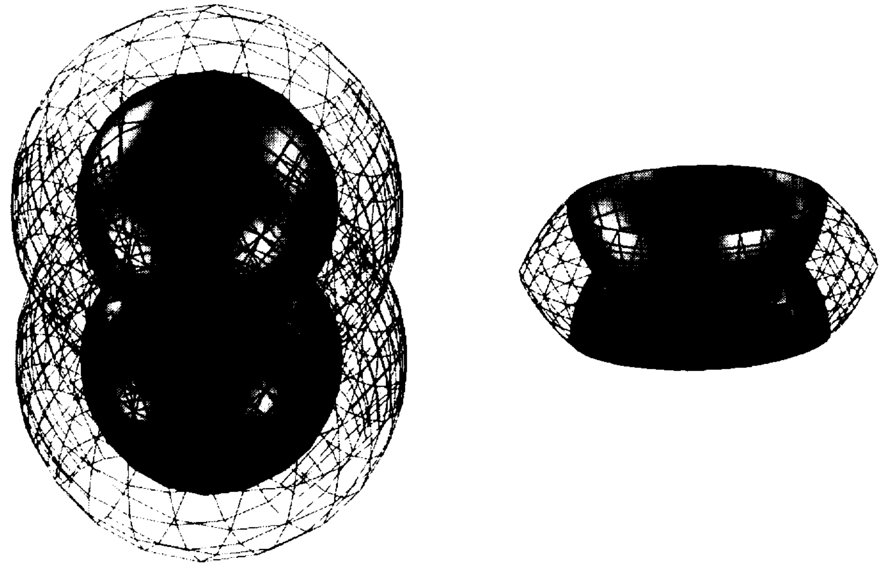

Figure 15: Collar grid between two intersecting spherical shells

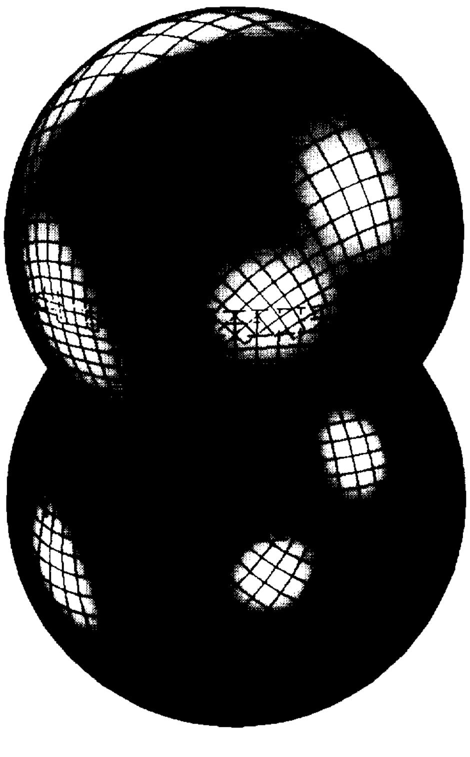

Figure 16: Overlapping grid for two intersecting spheres

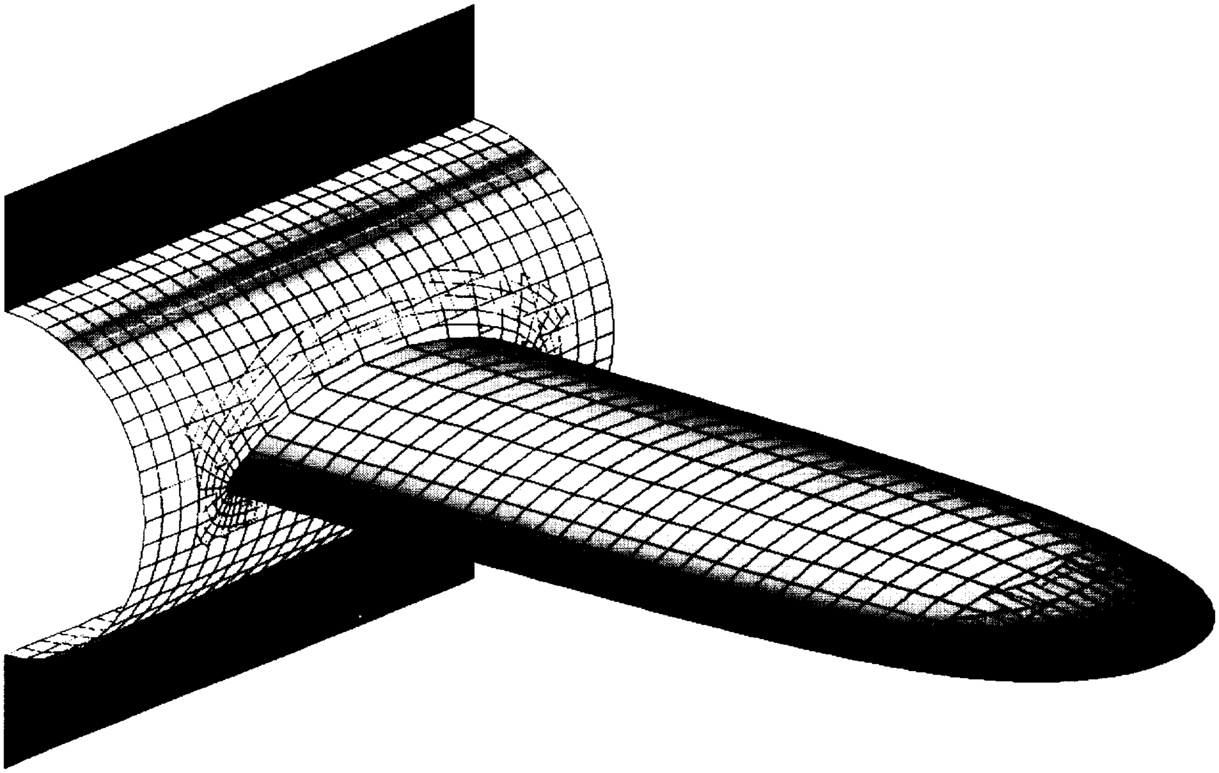

Figure 17: Overlapping grid created using intersecting surfaces